大型云存储中心分析与设计

2015211527 罗暄澍

目录

概述

在大数据时代，网络负载、速度方面的大幅提高，使得利用率高效、基础设施廉价、可以按需使用的云计算服务成为可能。而云存储是云计算概念上延伸，是其一种应用模式。它通过集群应用、网络技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能。公有云存储是云存储的主要实现模式。公有云存储服务主要由一些大公司来提供。例如 Amazon 的 AWS，微软的 Azure 等等。国内的服务提供商主要有阿里云、百度云等服务商。本文将对阿里云的存储架构、关键技术和应用适用类型进行分析，并在此基础上，设计一个可服务于全国性在线考试的大型云存储中心。该存储系统将全国划分为七个大区，而每个大区的数据中心采用服务器集群的方式进行数据存储。针对不同的存储对象，本文设置了不同的存储策略。针对安全性和可靠性，抛开产品本身的特性，提出了一些对管理员的操作建议、对系统设计的改进和面对不同问题防护措施。

一、阿里云的存储架构、关键技术和应用使用类型

阿里云是国内领先的云计算及人工智能服务供应商。区别于自建服务器存储，阿里云旗下成熟的、正式投放市场的云存储产品架构有以下 5 种：对象存储 OSS、块存储、文件存储 NAS、表格存储 TableStore 和归档存储 OAS。下面将逐个对五种产品进行分析。

(一) 对象存储 OSS

阿里云对象存储服务（Object Storage Service，简称 OSS），是阿里云提供的海量、安全、低成本、高可靠的云存储服务。它适合存储一些静态文件或对静态文件做一些简单的处理。比如网站中的图像、配置文件、视频、安装包等等。用户可以通过 HTTP 直链进行访问或下载。

1、关键技术

(1) 具有与平台无关的 RESTful API 接口

(2) 针对每一个文件都能提供一个 URL 直链，并且每个文件有两套备份
(3) 可用 HTTP 协议进行访问，并且支持 HTTPS 加密
(4) 支持多线 BGP 网络、辅以 CDN 进行内容分发，可以获得高响应速度

2、应用使用类型

OSS 对象存储是一个分布式的对象存储服务，提供的是一个 Key-Value 对形式的对象存储服务。用户可以根据 Object 的名称（Key）唯一的获取该 Object 的内容。虽然用户可以使用类似 test1/test.jpg 的名字，但是这并不表示用户的 Object 是保存在 test1 目录下面的。对于 OSS 来说，test1/test.jpg 仅仅只是一个字符串，和 a.jpg 这种并没有本质的区别。因此不同名称的 Object 之间的访问消耗的资源是类似的。

而文件系统是一种典型的树状索引结构，一个名为 test1/test.jpg 的文件，访问过程需要先访问到 test1 这个目录，然后再在该目录下查找名为 test.jpg 的文件。因此文件系统可以很轻易的支持文件夹的操作，比如重命名目录、删除目录、移动目录等，因为这些操作仅仅只是针对目录节点的操作。这种组织结构也决定了文件系统访问越深的目录消耗的资源也越大，操作拥有很多文件的目录也会非常慢。

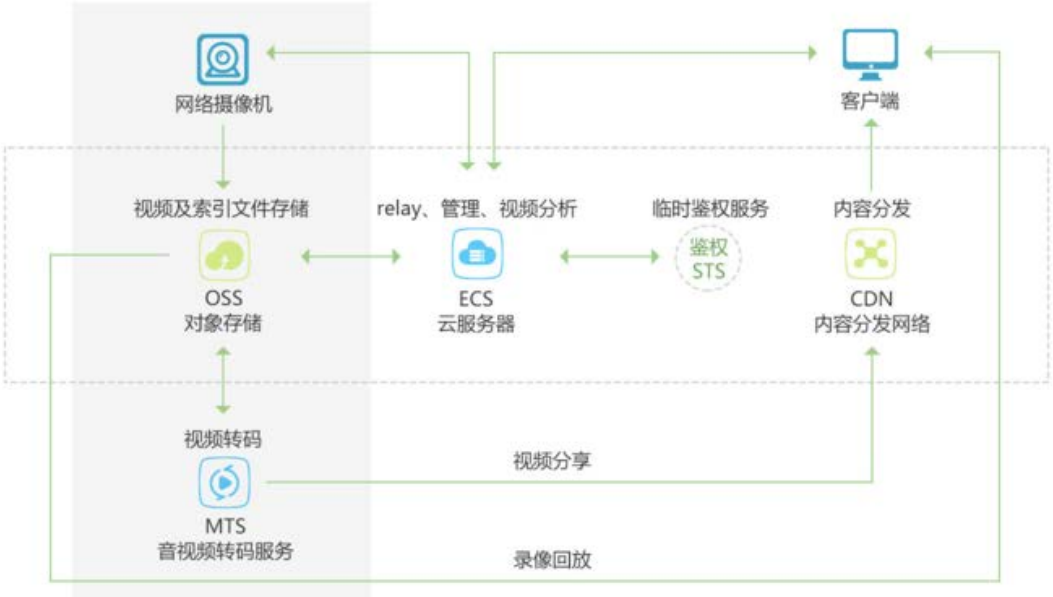
OSS 保存的 Object 不支持修改（追加写 Object 需要调用特定的接口，生成的 Object 也和正常上传的 Object 类型上有差别）。用户哪怕是仅仅需要修改一个字节也需要重新上传整个 Object。而文件系统的文件支持修改，比如修改指定偏移位置的内容、截断文件尾部等，这些特点也使得文件系统拥有广泛的适用性。但另外一方面，OSS 能支持海量的用户并发访问，而文件系统会受限于单个设备的性能。

因此，将 OSS 映射为文件系统是非常低效的。如果一定要挂载成文件系统的话，建议尽量只做写新文件、删除文件、读取文件这几种操作。使用 OSS 应该充分发挥其优点，即海量数据处理能力，优先用来存储海量的非结构化数据，比如图片、视频、文档等。

基于上述分析，我认为 OSS 对象存储适用于以下几种情况：

(1) 图片和音视频等应用的海量存储(例：湖南卫视)

OSS 可用于图片、音视频、日志等海量文件的存储。各种终端设备、Web 网站程序、移动应用可以直接向 OSS 写入或读取数据。湖南卫视的移动端和 PC 端的软件视频流就使用了 OSS 服务。它支持流式写入和文件写入两种方式，如下图所示：



(2) 网页或者移动应用的静态和动态资源分离(例：今日头条)

利用 BGP 带宽，OSS 可以实现超低延时的数据直接下载。也可以配合阿里云 CDN 加速服务，为图片、音视频、移动应用的更新分发提供最佳体验。今日头条客户端使用 OSS 和云服务器配合的模式，实现静态数据的利用和维护。如下图所示：

(3) 云端数据处理(例：新浪微博)

上传文件到 OSS 后，可以配合媒体转码服务(MTS)和图片处理服务（IMG）进行云端的数据处理，如下图所示：

(二) 块存储

块存储是为云服务器 ECS 提供的低时延、持久性、高可靠的数据块级随机存储。块存储支持在可用区内自动复制数据，防止意外硬件故障导致的数据不可用。支持对挂载到 ECS 实例上的块存储做分区、创建文件系统等操作，并对数据持久化存储。

1、关键技术

块存储由于使用了高性能的 HDD 或 SSD 甚至是 NVMe SSD 作为存储介质，所以具有良好的读写性能。

(1) 使用了分布式多副本技术，可以提供高效的数据随机访问能力，避免硬件故障带来的数据丢失问题。这点与 RAID 1 的效果相似。
(2) 使用了快照技术，可以对某一时间点的某一个磁盘进行数据备份，防止因管理员误操作或者病毒攻击带来的不可挽回的损失。我认为该功能是一个更高级的备份功能，可保存数据的历史版本，配合分布式多副本技术，可保证数据的万无一失。并可支持头东创建副本
(3) 具有弹性扩容能力。已有云盘支持灵活扩容，利用存量快照来创建块存储时，可为块存储配置大于快照容量的存储空间，既可以满足块存储以快照数据副本开始工作的诉求、又可以提供更大的存储容量。
(4) 数据加密功能。这个加密功能过程是静默的、对业务流程没有任何影响。管理员也不需要增加任何操作步骤。

2、应用使用类型

(1) SSD 云盘(贵、高 IOPS)

由于具有高性能，所以适合于 I/O 密集型应用、中大型关系数据库或者 NoSQL 数据库的建立。

(2) 高效云盘(便宜、低 IOPS)

性能的制约，导致其仅适合于创建开发与测试业务、小型负载数据库或系统盘。

选择 SSD 云盘还是高效云盘，需要管理员考虑产品需求和性价比，选取合适的方案

(三) 文件存储 NAS

Network Attached Storage(简称 NAS）在课上已经有所了解。阿里云 NAS 文件存储架构是面向 ECS 实例、HPC 和 Docker 的文件存储服务，提供标准的文件访问协议，用户无需对现有应用做任何修改，即可使用具备无限容量及性能扩展、单一命名空间、多共享、高可靠和高可用等特性的分布式文件系统。

也就是说阿里云的文件存储 NAS 架构相对于传统的 NAS，采用了分布式存储，使得它不存在传统 NAS 的“单点问题”。支持弹性扩展，使得阿里云 NAS 存储容量和吞吐能力都可以简单升级，而传统 NAS 需要重新购买设备进行数据转移。

1、关键技术

相对于传统 NAS，阿里云 NAS 具有以下额外的技术：

(1) 共享访问支持。多个主机实例可以同时访问一个 NAS 实例，适合跨多个主机部署的应用程序访问相同数据来源的应用场景。
(2) 弹性扩容。使得 NAS 的存储容量和吞吐能力都可以简单地进行升级或者为了节省成本进行降级。

2、应用使用类型

阿里云 NAS 有以下 5 点功能特性：

- 无缝集成：支持 NFSv3 及 NFSv4 协议，使用标准的文件系统语义访问数据，主流的应用程序及工作负载无需任何修改即可无缝配合使用。
- 共享访问 :多个计算节点可以同时访问同一个文件系统实例，非常适合跨多个 ECS、HPC 或 Docker 实例部署的应用程序访问相同数据来源的应用场景。
- 弹性伸缩 :单文件系统容量上限 10PB，按实际使用量付费，充分满足弹性伸缩需求。
- 安全控制：通过网络隔离（专有网络）/用户隔离（经典网络）、文件系统标准权限控制、权限组访问控制和 RAM 主子账号授权等多种安全机制，保证文件系统数据安全

万无一失。

- 线性扩展的性能：可为应用工作负载提供高吞吐量与高 IOPS、低时延的存储性能，同时性能与容量成线性关系，可满足业务增长需要更多容量与存储性能的诉求。

基于这些功能特性，阿里云 NAS 适用于以下情景：

(1) 服务器日志共享

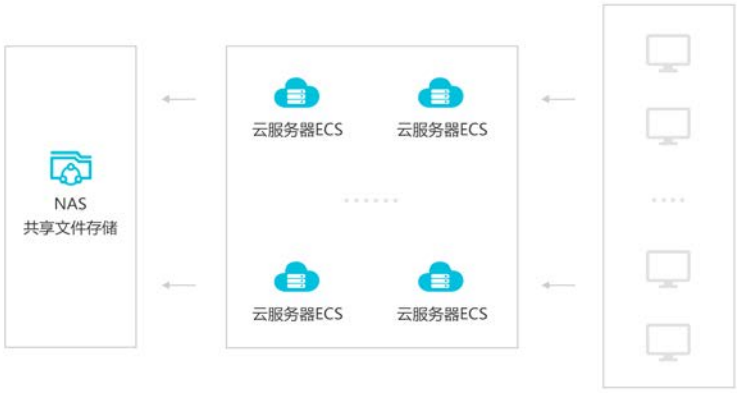
将多个 ECS 实例上的应用服务器日志存放在共享的文件存储上，方便后续的日志集中处理与分析

(2) 负载均衡服务器高可用

负载均衡 SLB 连接多个 ECS 实例场景，这些 ECS 实例上的应用将数据存放在共享的文件存储上，以实现负载均衡服务器的高可用

(3) 企业办公文件共享

企业员工办公需要访问和共享相同的数据集，管理员可创建 NAS 文件系统，为组织中的个人提供数据访问，并可设置文件或目录级别的用户和用户组权限



(四) 表格存储 TableStore

表格存储（Table Store）是构建在阿里云飞天分布式系统之上的分布式 NoSQL 数据存储服务。表格存储通过数据分片和负载均衡技术，实现数据规模与访问并发上的无缝扩展，提供海量结构化数据的存储和实时访问。

1、关键技术

(1) 与对象存储 OSS 一样，提供标准的、与平台无关的 RESTful API 接口。
(2) 使用分布式架构体系和大数据模型，可以自动实现负载均衡及热点迁移机制。数据存储及访问并发性能可以无限扩展
(3) 提供对大数据的支持。这种支持一定程度上是云计算与云存储的融合。支持 MaxCompute 直读直写以及 EMR Hadoop/Spark/Hive/Flink 等各类开源组件访问
(4) PK 自增列。它可以解决即时聊天系统、Feed 流等等的高并发访问的问题。

2、应用使用类型

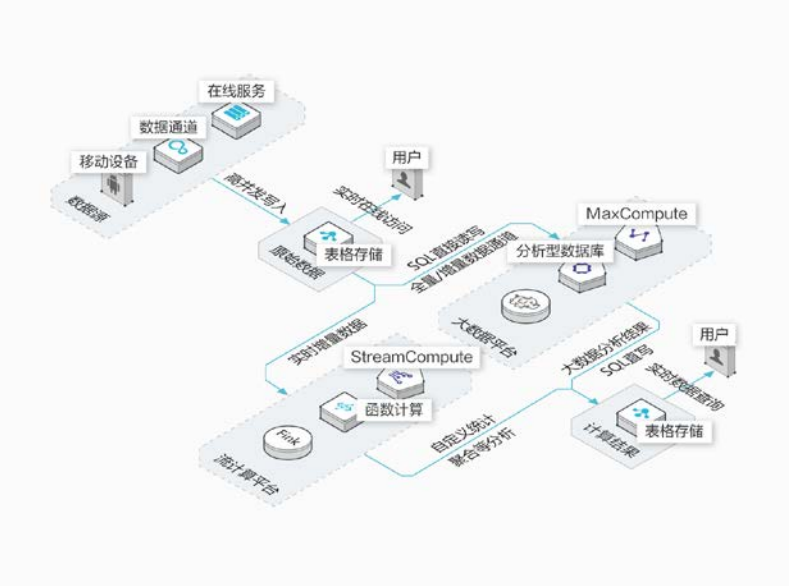
表格存储具有以下产品优势：

- 扩展性：
  - 动态调整预留读/写吞吐量：在创建表的时候，应用程序可以根据业务访问的情况来配置预留读/写吞吐量。表格存储根据表的预留读/写吞吐量进行资源的调度和预留，从而获得更低的资源使用成本。在使用过程中，还可以根据应用程序情况动态修改预留读/写吞吐量。
  - 无限容量：表格存储中表的数据量没有上限，随着表数据量的不断增大，表格存储会进行数据分区的调整从而为该表配置更多的存储。
- 数据可靠性：表格存储将数据的多个备份存储在不同机架的不同机器上，并会在备份失效时进行快速恢复，提供了极高的数据可靠性。
- 高可用性：通过自动的故障检测和数据迁移，表格存储对应用程序屏蔽了机器和网络的硬件故障，提供了高可用性。
- 管理便捷：应用程序无需关心数据分区的管理、软硬件升级、配置更新、集群扩容等繁琐的运维任务。
- 访问安全性：表格存储对应用程序的每一次请求都进行身份认证和鉴权，以防止未授权的数据访问，确保数据访问的安全性。
- 强一致性：表格存储保证数据写入强一致，写操作一旦返回成功，应用程序就能立即读到最新的数据。
- 灵活的数据模型：表格存储的表无固定格式要求，每行的列数及不同行同名列的类型可以不相同，支持多种数据类型，如 Integer、Boolean、Double、String 和 Binary。
- 按量付费：表格存储根据用户预留和实际使用的资源进行收费，起步门槛低。
- 监控集成：用户可以从表格存储控制台实时获取每秒请求数、平均响应延时等监控信息。

基于上述分析，表格存储适用于以下几种应用类型：

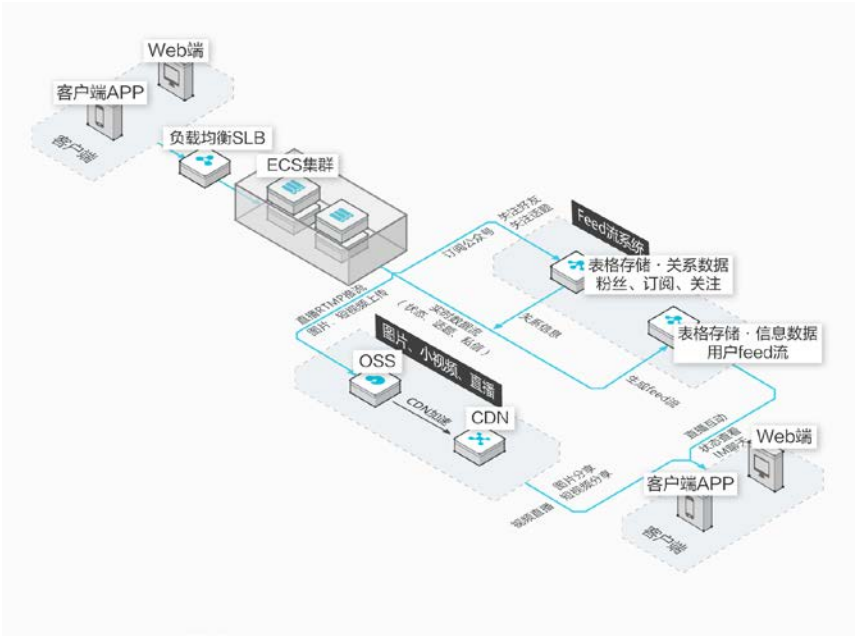
(1) 大数据存储与分析

表格存储提供低成本、高并发、低延时的海量数据存储与在线访问，提供增量以及全量数据通道并支持 MaxCompute 等大数据分析平台的 SQL 读写，高效的增量流式读接口可对数据进行实时流计算。



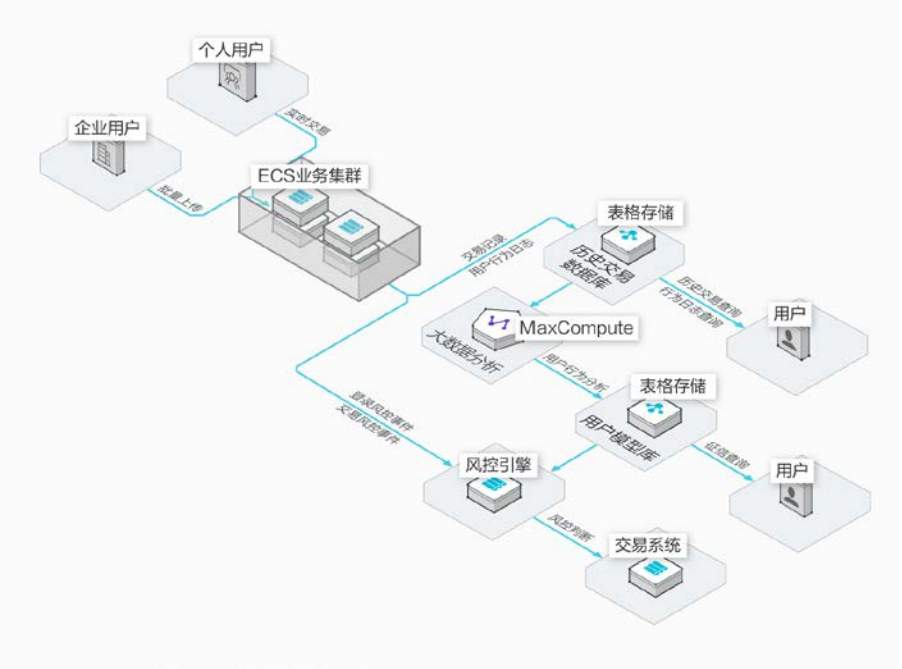
(2) 互联网社交 Feed 流(例：钉钉)

使用表格存储来存储大量的 IM 聊天、以及评论、跟帖和点赞等社交 Feed 流信息，表格存储的弹性资源按量付费能够以较低的成本满足访问波动明显、大并发低延时的需要。钉钉客户端数据中，Feed 流中的数据即非常适合使用表格存储，因为其数据具有明显的结构。钉钉客户端的组织框图如下：



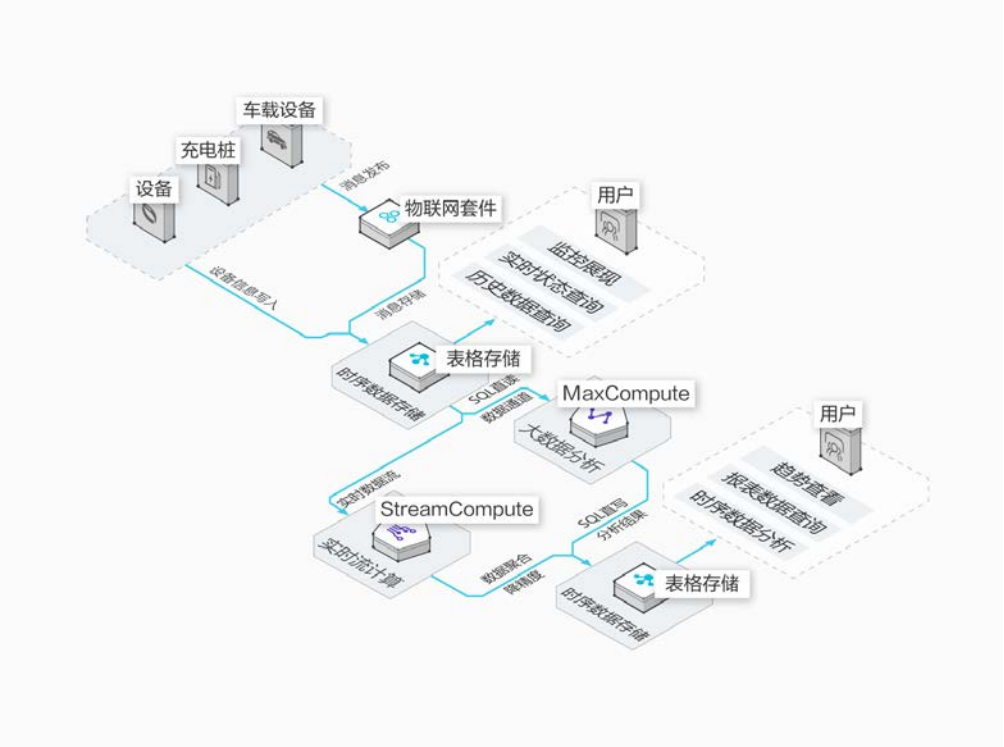
(3) 金融交易与风控

低延时、高并发，弹性存储可以使风控系统永远工作在较好的状态，并可控制交易风险。数据结构也可以灵活变动，能够让业务模式跟随市场需求快速迭代。



(4) 物联网时序数据(例：施耐德电气)

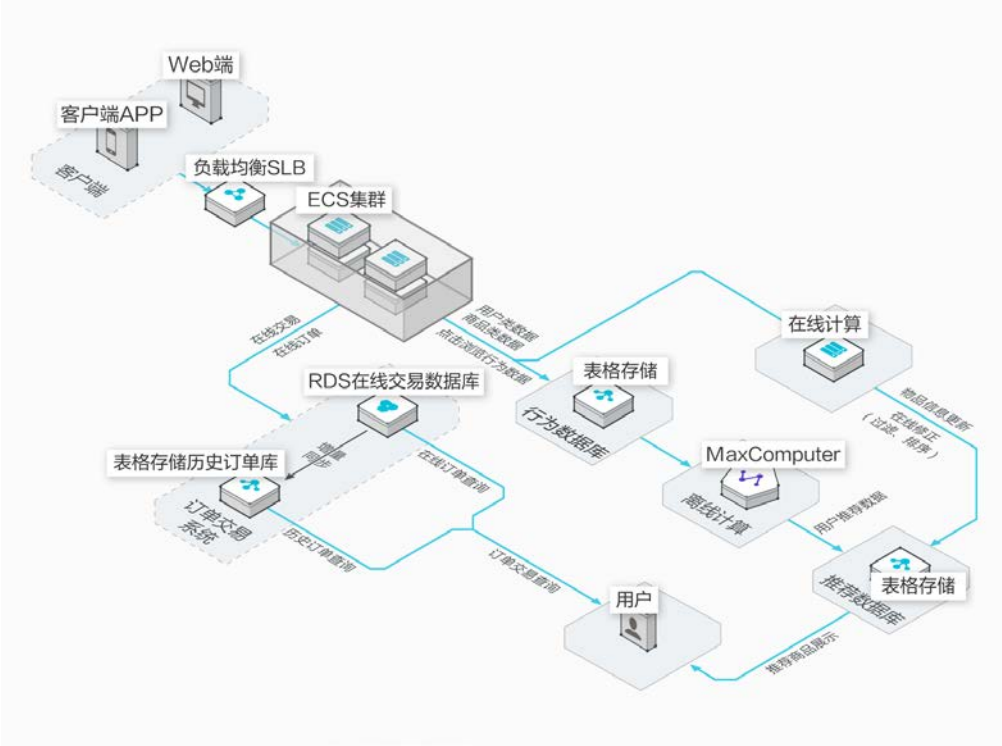
表格存储的高性能可以满足 IoT 设备、监控系统等时序数据的存储需求，大数据分析 SQL 直读以及高效的增量流式读接口可让数据完成离线分析和实时流计算。施耐德电气作为老牌的重工业和工业控制公司，也在大力推进智能硬件。智能硬件的数据上传到云端后，可以使用表格存储，并搭配 SQL 数据库操作进行数据分析，再次利用表格存储，将分析结果保存，以便推送回用户终端。流程图如下：



(6) 电商订单与广告推荐(例：淘宝、天猫)

使用表格存储大量的历史交易订单使得数据规模与访问性能相对不受限制，配合大数据

计算服务，实现精准营销。弹性资源和按量付费，可应对所有用户在线高峰时刻。



(五) 归档存储 OAS

可提供低成本、高可靠的数据归档服务，适合于海量数据的长期归档、低成本存储（几个月、几年乃至几十年）和备份。价格十分便宜，但是相应的，响应速度非常慢。所以顾名思义，仅仅适合数据的归档。

1、关键技术

(1) 归档的每 MB 数据都会有指纹保存。保证了数据的安全性。
(2) 支持弹性扩展
(3) 提供 Restful 的接口支持，与平台无关，轻松访问数据
(4) 不需要中转数据即可在 OSS 对象存储实例与 OAS 归档存储实例之间进行数据传输。这是阿里出于对自己公司产品易用性方面的考虑。

2、应用使用类型

归档存储最大的优势就是低成本，所以它适用于以下两种情况：
(1) 低频低成本存储
适用于存储用户存储后很少访问的数据，或者用户提取数据时能够容忍 0-4 小时的时间延迟时，可以使用此种类型的应用
(2) 长期备份与归档
适用于用户的数据需要长期备份(如存储几个月，乃至几年)，或者替换用户自己搭建的磁带库，而使用云归档进行存储的情况。

(六) 产品应用范围和价格比较

通过以上资料查询和架构特点归纳，以及阿里云官网对这五种产品的定价，我认为可用如下表格进行总结：

	特点及适用范围	价格(1-5 递增)
对象存储 OSS	适合包括图片、音视频的多种**静态媒体**的保存	2
块存储	高响应、低延迟、静默加密，适用于**大量读写**	3
文件存储 NAS	适用于共享文件存储	4
表格存储 TableStore	**结构化**、可与云计算融合，适用于**高并发访问**	5
归档存储 OAS	适用于文件历史版本的存储、**长期备份**	1

二、设计一个可服务于全国性在线考试的大型云存储中心

要求：

基于上述分析，设计一个大型云存储中心，该中心服务于全国性在线考试系统(包括客观和主观试题，其中主观试题要求采用非结构化数据存储方式)，系统要求考生全国地理位置非均匀分布、在线考试考生十万以上，给出总体设计框架、采用的关键技术和管理技术，要求有针对性的提出海量数据存储策略、网络存储架构、可靠性保障策略和安全保障策略。

(一) 前期分析

1、地理因素分析与架构概述

根据 2017 年第一季度人口统计数据，我们可得如下统计结果：

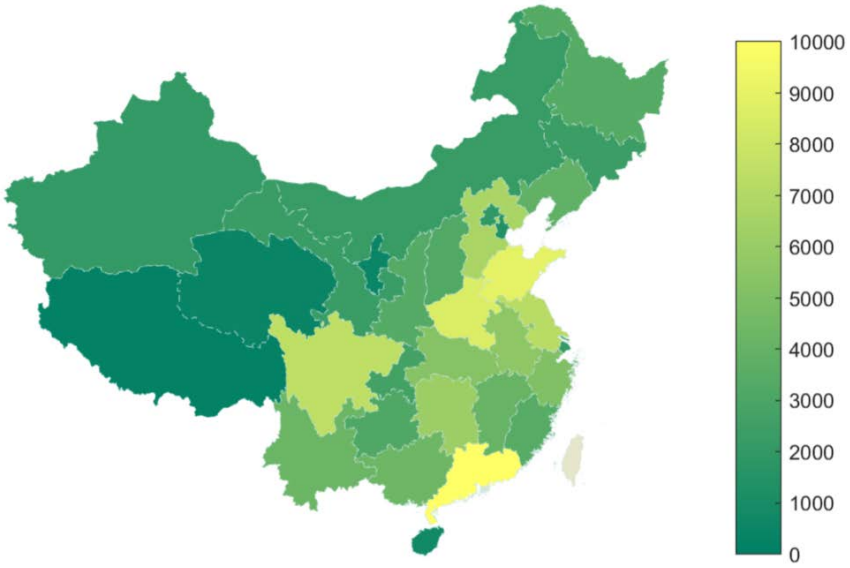


Figure 1 2017 年第一季度我国人口密度图(单位：万人)

一个全国性的大型云存储中心，需要采用分布式策略，而非单中心，来保证访问可靠性，一定程度上环节高并发的问题。根据 **Figure 1** 的人口密度图，我们可以假设各省参加考试的人数与人口数量成正比。综合考虑当地经济水平(需要专业人员进行维护)，我认为应该在

沈阳(东北)、北京(华北 1)、济南(华北 2)、上海(华东)、成都(西南)、西安(西北)、广州(华南)七个城市设立分布式数据库。

每个地方的数据库之间要定时进行数据同步。而每个数据库也不应使用中心服务器，而是使用中心机组，利用服务器集群来解决稀释高并发，从而获得高可用性。

2、不同时间段的试题存储方案

在不考试的阶段，此系统主要负责的应该是以往试题的存储、题库的存储和少量的题目改动工作。这种情况下，用户终端提交的请求(例：修改试题内容等)完全可以接受长时间的延迟。同时为了节约成本，选择阿里云 OAS 归档存储最为合适。同时，OAS 归档存储可以非常简单地进行实例转换，转化为 OSS 对象存储等等其他阿里云云存储产品。所以在非考试时段使用 OAS 归档存储十分合适。

3、不同试题的存储方案

首先确定的是在考试的时间段内，不可使用归档存储 OAS

(1) 客观试题

客观试题在要求中并没有给出特殊要求。我认为客观题可视为是结构化数据。客观题可被分为选择题、客观填空题和判断题。我们可以很清晰地写出其数据结构(类 Java)。

```
//选择题
public class choiceQuestion extends question{
    private String id;          //题号
    private String quiz;        //问题
    private int ans;            //答案

    public boolean isRight(int stuAns) {...}
}

//客观填空
public class filBlankQuestion extends question{
    private String id;          //题号
    private String quiz;        //问题
    private String[] ans;       //答案

    public boolean isRight(String[] stuAns) {...}
}

//判断题
public class trueorflaseQuestion extends question{
    private String id;          //题号
    private String quiz;        //问题
    private boolean ans;        //答案

    public boolean isRight(boolean stuAns) {...}
}
```

同时，这种客观试题亦可以进行机器阅卷，免去人工过程。所以理想的服务器是在满足高并发的情况下，支持对数据的简单处理。而相应的，对数据上传速度并没有太高的要求，只需保证数据传输正确即可。

所以，这里我们选择阿里云 OSS 对象存储比较合适。无论是表格存储还是块存储，对于客观题的评判都显得大材小用。阿里云 NAS 虽然是共享文件存储，看起来很符合这种情况，但是考虑到成本问题，和并非那么高的并发性(在线考试十万以上)，我认为 OSS 对象存储更为合适。

(2) 主观试题

主观试题要求采用非结构化数据存储方式。这样就导致表格存储并不适合，因为它仅支持结构化数据。同时考虑到主观试题的内容比较庞大，可以将试题内容和学生的答案分开存放。将试题内容放在对象存储 OSS 中，而将学生上传的答案放在块存储实例中。这样做同时节约了试题内容存储部分的成本。

另外，主观试题需要人工阅卷，这样在线系统要承担大量的读写文件操作，这种情况下阿里云 NAS 无论从性能还是成本上考虑都不适合，使用块存储无疑是更好的方案

(二) 架构设计

经过以上分析，我们将该在线系统存储策略分为以下几种情况并对应每种情况进行以下存储方式：

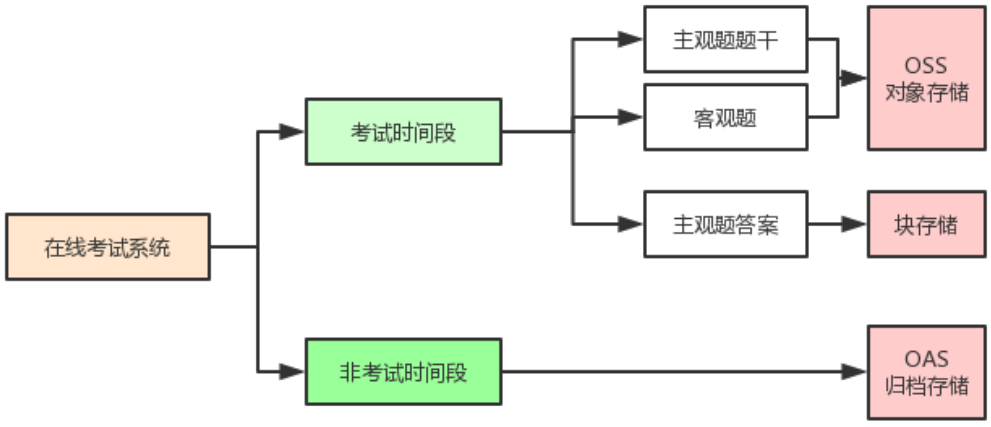


Figure 2 在线考试系统的存储策略

下面我们将分为：考试时间主观题题干、考试时间客观题、考试时间主观题答案、非考试时间段四个部分进行设计。

1、考试时间主观题题干

主观题的题干需要保证支持高并发下载。这里选择 OSS 对象存储，是考虑到成本，也考虑到题干均为静态多媒体内容，仅供下载。同时这部分内容不需要与用户之间发生任何交互。因此针对题干的架构十分简单。

在数据传输方面，我们可以借助 CDN 内容分发网络使考生可以从最近的几个服务器获取到试题内容，减少了延迟，提高了可靠性。云服务器在这里的作用是对考生请求进行处理。

整个流程可用下图来表示：

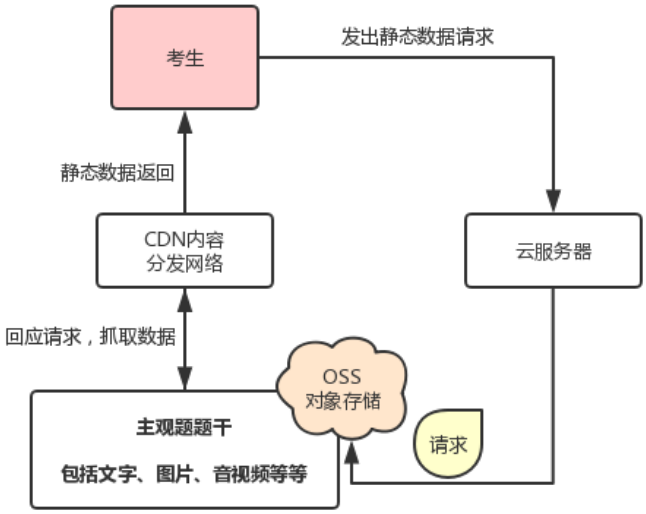


Figure 3 主观题题干存储策略

2、考试时间客观题

客观题本身仍为静态内容。但是为了节省成本，本文假设使用机器判卷，此时需要云服务器不仅要完成用户的静态数据请求，更要负责处理批卷子的过程，此过程可有一定的延迟。云服务器仍然以返回静态数据为主要的任务。同时，也要添加 CDN 进行内容分发。

所以针对考试时间的客观题存储和处理如下：

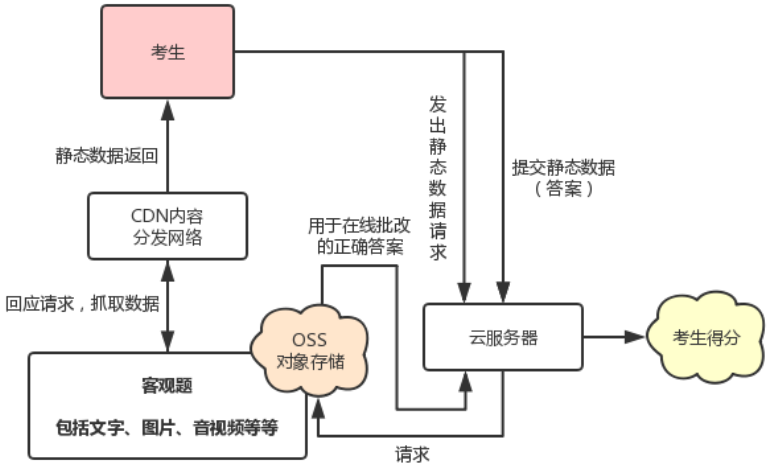


Figure 4 客观题存储策略

3、考试时间主观题答案

主观题答案在考试过程中会被大量学生同时上传，批卷子使可能也涉及到老师的在线读取或者下载之后离线判卷。这需要云存储架构具有高响应、低延迟的特性，最重要的使要适用于大量读写操作。所以这里使用块存储。同样，因为全国有七个服务器同时进行服务，所以增加 CDN 进行内容分发可以获得更好的效果。总结起来存储架构如下：

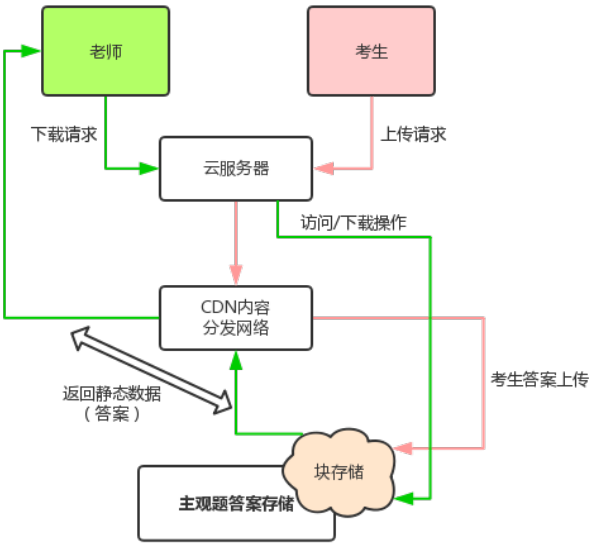


Figure 5 主观题答案存储策略

4、非考试时间

非考试时间，存储中心系统只涉及到存储题库、历年试题和修改试题等操作。这些操作用户都可以接受长响应时间，并且系统闲时的成本必须要有所限制。所以采用 OAS 归档存储最适合。而且 OAS 归档存储可以直接由管理员转化为阿里云的其他存储产品，使得系统运作十分方便。此种存储结构十分简单，可由下图表示：

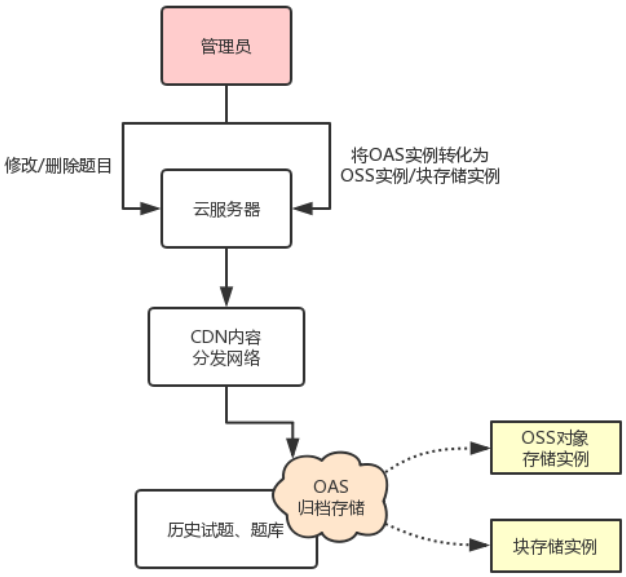
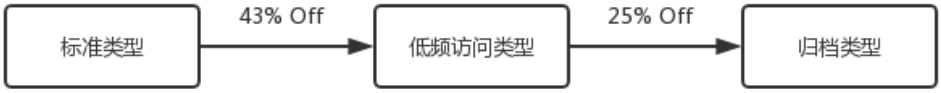


Figure 6 非考试时间存储策略示意

5、其他

OSS 对象存储的成本节约问题

OSS 对象存储本身仍然有以下三种分类。



为了保证在考试当天存储中心可以正常运转，管理员通常会提前将 OAS 归档存储实例转换为 OSS 对象存储实例进行测试。测试过程中的成本可以通过转化为低频访问类型或者归档类型的 OSS 对象存储实例控制。局部服务器测试通过后，再转化为标准类型进行全国系统测试。这样可以节约一部分成本。

(三) 可靠性与安全性

阿里云作为一个投放到市场的产品，一定会考虑到很多策略。作为使用者，我们要做的更多的是数据同步、架构设计、异常事件处理、时间的合理安排和文件安全的问题。

1、可靠性保障策略

(1) 数据一致性

对于一个考试系统，数据一致性主要分两个方面。一是保证服务器存储的数据，是学生最后一次提交的版本；二是在系统增加、修改、删除题目的时候，七个分布式服务器之间能够保持同步。

在分布式领域，CAP 理论高度概括了其技术目标。CAP 三者分别指：

1) **Consistency**： 一致性，数据一致更新，所有数据变动都是同步的
2) **Availability**：可用性，好的响应性能
3) **Partition tolerance**：分区容错性

CAP 理论指出，任何分布式系统只可同时满足 CAP 中的两点，而无法三者兼顾。首先最后一点，P 是无论如何都要保证的，因为 P 代表分布式系统在遇到任何网络分区故障的时候，仍然能够保证对外提供满足一致性和可用性的服务，除非是整个网络环境都发生了故障。这正是分布式系统建立要达到的主要目标之一。那么如果不想牺牲一致性，CAP 理论告诉我们只能放弃可用性，这显然是不可接受的。

因此，我们需要退而求其次，在保证 A 和 P 的情况下，尽可能提高 C(一致性)。数据一致性有三种策略去实现：

1) **强一致性**

当更新操作完成之后，任何多个后续进程或者线程的访问都会返回最新的更新过的值。这种是对用户最友好的。用户上一次写什么，下一次就保证能读到什么。根据 CAP 理论，这种实现需要牺牲可用性，显然是不可取的。

2) **弱一致性**

系统并不保证续进程或者线程的访问都会返回最新的更新过的值。系统在数据写入

成功之后，不承诺立即可以读到最新写入的值，也不会具体的承诺多久之后可以读到。这对于一个可用系统来说也是不可接受的。

3) 最终一致性

它是弱一致性的特定形式。系统保证在没有后续更新的前提下，系统最终返回上一次更新操作的值。在没有故障发生的前提下，不一致窗口的时间主要受通信延迟，系统负载和复制副本的个数影响。DNS 就是一个典型的最终一致性系统。

显然，在 CAP 理论的约束下，保证可用性和分区容错性的基础上，完成最终一致性是一种可行的方案。这样理想的系统将会退化为 BASE 理论的实例。其核心思想是即使无法做到强一致性，但每个应用都可以根据自身的业务特点，采用适当的方式来使系统达到最终一致性。BASE 分别指 Basically Available(基本可用)、Soft state(软状态)和 Eventually consistent(最终一致性)。那么对于全国在线考试系统来说：

1) 针对第一个目标，即保证系统存储的答案版本是学生上传的最终版本，我认为只要将学生准考证号设定为唯一主键，未提交答案的时候，答案存储区为空；提交答案后，学生如有更新，系统可以根据主键来寻找并更新答案，更新成功后发送一个 ACK。而学生端软件则应该对 ACK 进行超时判断，超过一定时间未收到则自动判定为未成功。此时系统将根据网络负载，选择再次提交或者更换到其他大区的服务器进行提交，以保证学生端能够在可接受的时延下完成资料的传输。
2) 针对第二个目标，即保证在非考试时间的系统闲时，管理员对试题的增加、修改、删除，在各个大区之间信息可以同步。即七个中心机组中任意一个对题库数据进行增加、修改或删除之后，其余六个也能进行同步。而这就涉及到分布式事务的问题。如果想让分布式部署的多台机器中的数据保持一致性，那么就要保证在所有节点的数据写操作，要不全部都执行，要么全部的都不执行。但是，一台机器在执行本地事务的时候无法知道其他机器中的本地事务的执行结果。所以他也就不知道本次事务到底应该 commit 还是 roolback。

为了解决分布式事务问题，通常有两类做法。

1) 引入一个“协调者”的组件来统一调度所有分布式节点的执行。引入“协调者”组件后，有两种可用的协议可以使用，分别为两阶段提交协议和三阶提交协议。
   - 二阶段提交(Two-phase Commit)的算法思路可以概括为：参与者将操作成败通知协调者，再由协调者根据所有参与者的反馈情报决定各参与者是否要提交操作还是中止操作。
   - 三阶段提交（Three-phase Commit），也叫三阶段提交协议（Three-phase commit protocol），是二阶段提交（2PC）的改进版本。与两阶段提交不同的是，三阶段提交有两个改动点。
     a) 引入超时机制。同时在协调者和参与者中都引入超时机制。
     b) 在第一阶段和第二阶段中插入一个准备阶段。保证了在最后提交阶段之前各参与节点的状态是一致的。

   三阶段提交对二阶段的单点故障问题进行了改进，并能减少阻塞，所以若引入“协调者”，三阶段提交协议更合适
2) 使用 Paxos 算法。这是一种基于消息传递的数据一致性算法。这种算法比引入“协调者”更有效。管理员可以优先选择此方法。

以上两点显然是认为七个大区的数据中心是平级关系。但是考虑实际情况，出于人力成本和管理的角度，通常对题库的操作请求都是由其中某一服务器提交的，此时七个数据中心

之间是主副关系(比如北京数据中心为主中心，其余六个为副)。这种情况下，实现的思想仍然不需要改变，而对每个副中心数据同步的实现策略则可以删去主动提交请求的模块，简化设计。

(2) 高并发的处理

针对考试前期大量下载试卷、考试后期大量上传请求带来的高并发问题，我认为有如下解决方案。

1) 每个大区使用服务器集群的中心机组，而非单点数据库

这一点在前文有所叙述。使用服务器集群可以有效稀释每个服务器的负载，减轻数据中心的压力。同时使用集群，也可以有效降低成本。对于考试系统来说，最重要的是保证数据的同步和可以储存所有学生最后一次提交版本的答案。为此，整理服务器集群带来的时间损耗是微不足道的。

2) 系统支持分题下载和上传

试卷整体上传和下载在时间上必然会集中，会在短时间内消耗大量带宽，而答卷过程中系统则相对空闲，这是十分不合理的。所以试卷应该支持部分下载和上传。在客户端系统，学生可以选择题号进入题面，这时只需要下载一部分即可，可将短时间内的大量请求分散，有效减轻负载。同理，答案上传也可以分大题进行上传，而非整体上传，同样可以达到减轻负载的效果。同时这种操作也是为学生着想，以免到时间没有上传答案，获得不合理的分数。这种策略至少可以保证学生上传完所有已经回答好的问题。

(3) 服务器副本

可靠性还可以通过增加数据副本来提高。我认为在分布数据中心的周边大城市应当设立常驻的 OAS 归档存储。但是建立太多又会大幅增加成本。我认为应该在全国设立两个左右 OAS 归档存储实例为好。这两个实例同样与主服务系统进行数据同步管理，不过同步周期可以相对设定长一些。

对于华东和华南这两个人口密集的地区，可能在考试过程中会出现系统崩溃的情况。我认为在这两个地区应该设立主副两个服务器实例。同时，客观题的批改可以在服务器相对闲时进行，而不必即时批改，以减小服务器处理器负载。

(4) 分发同步

这里我们使用 Linux 环境下的 Sersync 框架，搭配 inotify 与 rsync 技术实现对服务器数据实时同步分发。其中 inotify 用于监控 Sersync 所在服务器上文件系统的事件变化，rsync 是目前广泛使用的本地及异地数据同步工具，其优点是只对变化的目录数据操作。

Sersync 项目的优点：

- 使用 C++编写，对 Linux 系统文件产生的临时文件和重复的文件操作会进行过滤，再结合 rsync 同步到时候，会减少网络资源，因此速度更快。
- Sersync 配置起来很简单
- 使用多线程进行同步，尤其在同步较大文件时，能够保证多个服务器实时保持同步状态。
- Sersync 自带出错处理机制，通过失败队列对出错的文件重新同步，如果扔失败，则每 10 个小时对同步失败的文件再重新同步。
- Sersync 自带 crontab 功能，只需在 xml 配置文件中开启，即可按预先的配置，隔一段时间整理同步一次。

- Sersync 自带 socket 与 http 的协议扩展，可以满足有特需要去掉公司二次开发。

为各分布式数据中心建立好环境后，无论是采用主从式还是并列式的结构，管理员都可以从任何一个数据中心发起试题分发指令。在此框架中，分发试题的指令会生成一个 inotify 对象，rsync 技术会将该命令同步到各个数据中心。此时，被同步的数据中心应该给发起指令的数据中心回传一个 ACK 信号，并引入超时机制。所有数据中心准备好后，约定在考试开始时间（此信息也可作为 inotify 对象被同步）将试卷内容公开，实现试卷的同步分发。

2、安全保障策略

安全性策略主要考虑考试时的海量访问攻击和机密文件保护的问题

(1) CC/DoS 攻击的防御

它们的实现方式不同，但是主要思想类似，都是通过模拟多个用户不停的进行访问，使得被攻击的主机无法完成正常用户的请求(拒绝服务)。云服务提供商会对这些情况进行一定的防护，管理员也可以安装防护软件，我们这里更多考虑的是架构的优化和存储策略的改进。为了解决这个问题，我认为应该采取以下策略：

a) 这些攻击通常都是利用 TCP/IP 协议漏洞来攻击，所以在协议支持自定义的范围，我们可以进行如下修改：
   - 关闭不必要的服务(端口)
   - 限制同时打开的 SYN 半连接数目
   - 缩短 SYN 半连接的 TIME-OUT 时间
   - 尽量避免 NAT 的使用

b) 采用分布式组网、负载均衡、提升系统容量等可靠性措施，增强总体服务能力。对于我们的系统来说，服务器集群再次显现其优势。几个服务器的失效并不能影响整体的服务质量；

c) 系统要及时更新打补丁；

d) 尽量多的使用静态内容。比如考试数据存储中心里的批改系统，可以完全放到别的主机上来做；

e) 认真对代理请求进行识别。据统计，80%使用代理访问网站的行为是恶意的。系统要对代理请求格外注意，或者拒绝代理访问。

(2) 机密文件的防护

这一部分的攻击主要分为三种。

- 信息的泄露
- 信息的篡改
- 信息的故意毁坏(非物理损坏)

a) **针对信息泄露**

可以使用加密算法来提高安全性。这样即使攻击方获取到信息，也是经过加密的，无法解读内容。而对于文字加密和图像加密，可以使用不同的算法。这里要考虑算法的时间复杂度，解密过程要尽量快，而且最好安排在学生答卷子的客户端，本地解密；而上传的答案也要本地加密后再上传，保证数据在整个传输过程中都是密文。

b) **针对信息篡改**
   - 使用证书认证技术。有的认证需要第三方服务的参与，这种技术的安全性比较

高，但是会带来额外的成本，这需要考试的组织者根据需求来决定。

- 使用数字签名技术，它模拟了传统的对于数据文档的签名方式，是一种基于哈希的公钥或者私钥技术。简单的可以通过信息摘要来实现：比如 SHA/MD5 校验。

使用这些技术都可以及时发现数据被篡改的情况，从而做出应对措施。比如更新题库、重新出题等等。

c) **针对信息故意毁坏**

上文我们提到过服务器副本的设置。如果信息遭到严重破坏，可以从 OAS 备份对象中恢复出数据。同时要根据日志记录内容或者其他手段收集证据，及时报警。

以上做法是信息泄露或者信息篡改之后的亡羊补牢之举，姑且可以算作是应用层的解决方案。在物理层和数据链路层上，可以建立专门的传输信道，在底层扼杀泄露的可能。但是这个需要有关部门的许可，成本也是个大问题。所以在物理层上应该是无法解决的。

除此之外就是在网络层。信息泄露、信息篡改的过程实现以及信息的故意破坏都是在网络层执行。这一部分可以交给防火墙和管理员对协议参数的自定义。防火墙作为计算机内部与外部连接通信的一道屏障，会对于外部发来的信息进行拦截，并且对于可疑的数据包进行防护，拦截。这一部分同样可以部分参考 CC/DoS 防御中的内容。

(3) 数据中心灾难的应对

可以通过建立灾备中心来解决这一问题。灾备中心的选址很重要。在中国应该尽量选择内陆的非地震带地区。通过查阅资料可知，我国内蒙古自治区和贵州省适合建立灾备中心（例：苹果公司的灾备中心选址在贵州）。一旦遇到数据中心灾难，根据灾难类型，管理员选择冷备、暖备/热备、双活三种顶层恢复模式之一进行恢复。通过查询资料得知，使用不同的数据恢复方法，面对不同的数据损坏程度，恢复时间从一分钟以内到小时级别都有可能。这需要管理员能够在最短的时间内分析出数据恢复的时间。如果达到小时级别，可能考试就要推迟。以下是常见的数据复制技术极其技术特点：

	基于存储的数据复制	虚拟存储技术	操作系统层数据复制	应用程序层数据复制
基本原理	数据的复制过程通过本地的存储系统和远端的存储系统之间的通信完成。	复制技术是伴随着存储局域网的出现引入的，通过构建虚拟存储上实现数据复制。	通过操作系统或者数据卷管理器来实现对数据的远程复制。	数据库的异地复制技术，通常采用日志复制功能，依靠本地和远程主机间的日志归档与传递来实现两端的数据一致。
平台要求	同构存储	与平台无关	同构主机、异构存储	与平台无关
复制性能	高	高	高	较高
资源占用	对生产系统存储性能有影响	对网络要求高	对生产系统主机性能有影响	占用部分生产系统数据库资源
成熟度	成熟	成熟度有待提高，非主流复制技术。	成熟	成熟

投入成本	高，需要同构存储	较高，需要专有设备	较高，需要同构主机	一般。部分软件免费，如DataGuard
复制软件	IBM PPRC EMC SRDF HDS TrueCopy	Brocade Tapestry DMM UIT SVM	IBM AIX LVM HP-UINX MirrorDisk Symantec SF/VVR	Oracle DataGuard Oracle GoldenGate DNT IDR

3、总结

可靠性与安全性方面，本文讨论更多的是抛开云服务厂商提供的安全套件以外的策略。比如数据中心内部采用服务器集群，试题内容和答案上传可以分块且本地加解密，全程密文传输等等。云服务提供商的保障也很重要，管理员要综合考虑来制定不同的策略。